

1 Claims 1, 9, 16, 23, 33 and 41-44 are amended.

2 Claims 2, 12, 20, 27 and 35 are cancelled without prejudice.

3 Claims 1, 3-11, 13-19, 21-26, 28-34 and 36-44 remain in the Application as
4 follows:

5
6 **1. (Currently Amended)** One or more tangible computer-readable
7 media comprising a flash memory driver that is executable by a computer to
8 interface between a file system and one or more flash memory media, the flash
9 memory driver comprising:

10 flash abstraction logic that is invokable by the file system to manage flash
11 memory operations without regard to the type of the one or more flash memory
12 media; and

13 flash media logic configured to interact with different types of the flash
14 memory media;

15 wherein the flash abstraction logic invokes the flash media logic to perform
16 memory operations that are potentially performed in different ways depending on
17 the type of the flash memory media, and further wherein the flash memory driver
18 is flash memory medium agnostic, and wherein one of the flash memory
19 operations includes performing wear-leveling operations associated with the flash
20 memory medium by way of circular and continuous advancement of a write
21 pointer.

22
23 **2. (Cancelled).**
24
25

1 **3. (Original)** The flash memory driver as recited in Claim 1,
2 wherein one of the flash memory operations includes maintaining data integrity of
3 the flash memory medium.

4
5 **4. (Original)** The flash memory driver as recited in Claim 1,
6 wherein one of the flash memory operations includes handling recovery of data
7 associated with the flash memory medium after a power-failure.

8
9 **5. (Original)** The flash memory driver as recited in Claim 1,
10 wherein one of the flash memory operations includes mapping status information
11 associated with physical sectors of the flash memory medium for use by the file
12 system.

13
14 **6. (Previously Presented)** The flash memory driver as recited in
15 Claim 1, wherein the flash medium logic is further configured to translate
16 commands received from the file system to physical sector commands for issuance
17 to the flash memory media.

18
19 **7. (Previously Presented)** The flash memory driver as recited in
20 Claim 1, wherein the flash medium logic is user programmable to read, write and
21 erase data to and from the flash memory media.

22
23 **8. (Previously Presented)** The flash memory driver as recited in
24 Claim 1, wherein the flash media logic is configured to perform error code
25 correction associated with the flash memory media.

1 **9. (Currently Amended)** A flash driver, comprising:
2 flash abstraction logic, interposed between a file system and a flash
3 memory medium, configured to:

4 (a) map a logical sector status from the file system to a physical sector
5 status of the flash memory medium; and

6 (b) maintain memory requirements associated with operating the flash
7 memory medium;

8 wherein the flash driver is located remote from the flash memory medium,
9 and wherein the memory requirements include managing wear-leveling operations
10 associated with the flash memory medium by way of circular and continuous
11 advancement of a write pointer.

12
13 **10. (Original)** The flash driver as recited in Claim 9, further
14 comprising a user programmable flash medium logic, configured to read, write
15 and erase data to and from the flash memory medium.

16
17 **11. (Original)** The flash driver as recited in Claim 9, further
18 comprising a user programmable flash medium logic configured to receive and
19 translate specific operational commands from the file system associated with
20 reading and writing data to the flash memory medium.

21
22 **12. (Cancelled).**
23
24
25

1 **13. (Original)** The flash driver as recited in Claim 9, wherein the
2 memory requirements include maintaining data integrity of the flash memory
3 medium.

4
5 **14. (Original)** The flash driver as recited in Claim 9, wherein the
6 memory requirements include handling recovery of data associated with flash
7 memory medium after a power-failure.

8
9 **15. (Original)** The flash driver as recited in Claim 9, further
10 comprising a flash medium logic, programmably configurable by a user to perform
11 error code correction associated with the flash memory medium.

12
13 **16. (Currently Amended)** A flash driver, comprising:
14 user programmable flash medium logic, configured to read, write and erase
15 data to and from a flash memory medium; and

16 flash abstraction logic, interposed between a file system and flash memory
17 medium to maintain universal requirements for the operation of the flash memory
18 medium;

19 wherein the flash memory driver is flash memory medium agnostic, and
20 wherein the universal requirements include managing wear-leveling operations
21 associated with the flash memory medium by way of circular and continuous
22 advancement of a write pointer.

1 **17. (Original)** The flash driver as recited in Claim 16, wherein the
2 flash abstraction logic passes specific commands associated with certain types of
3 flash memory media directly to the flash medium logic for translation and
4 execution.

5
6 **18. (Original)** The flash driver as recited in Claim 16, wherein the
7 flash abstraction logic is an interface between the flash medium logic and the file
8 system.

9
10 **19. (Original)** The flash driver as recited in Claim 16, wherein the
11 universal requirements include maintaining data integrity of the flash memory
12 medium.

13
14 **20. (Cancelled).**

15
16 **21. (Original)** The flash driver as recited in Claim 16, wherein the
17 universal requirements include handling recovery after a power-failure.

18
19 **22. (Original)** The flash driver as recited in Claim 16, wherein the
20 flash medium logic comprises a set of programmable entry points that can be
21 implemented by a user to interface with the type of flash memory medium
22 selected.

1 **23. (Currently Amended)** A processing device that uses a flash
2 memory medium for storage of data, comprising:

3 a file system, configured to control data storage for the processing device;
4 flash media logic, configured to perform physical sector operations to a flash
5 memory medium based on physical sector commands, wherein the flash medium
6 logic comprises a set of programmable entry points that can be implemented by a
7 user to interface with any type of flash memory medium selected; and

8 flash abstraction logic, configured to maintain flash memory requirements
9 that are necessary to operate the flash memory medium, wherein the flash memory
10 requirements include managing wear-leveling operations associated with the flash
11 memory medium by way of circular and continuous advancement of a write
12 pointer.

13
14 **24. (Original)** The processing device as recited in Claim 23, wherein
15 the flash abstraction logic passes physical logic commands associated with certain
16 types of flash memory medium directly to the flash memory medium logic for
17 translation and execution.

18
19 **25. (Original)** The processing device as recited in Claim 23, wherein
20 the flash abstraction logic is an interface between the flash medium logic and the
21 file system.

22
23 **26. (Original)** The processing device as recited in Claim 23, wherein
24 the flash memory requirements include maintaining data integrity of the flash
25 memory medium.

1 **27. (Cancelled).**

2
3 **28. (Original)** The processing device as recited in Claim 23, wherein
4 the flash memory requirements include handling recovery after a power-failure.

5
6 **29. (Original)** The processing device as recited in Claim 23, wherein
7 the requirements are common to a plurality of different flash memory media.

8
9 **30. (Original)** The processing device as recited in Claim 23, wherein
10 the flash medium logic comprises a set of programmable entry points that can be
11 implemented by a user to perform error code correction with the type of flash
12 memory medium used in the processing device.

13
14 **31. (Original)** The processing device as recited in Claim 23, whereby
15 the flash medium logic relieves the flash abstraction logic from performing
16 translation of the physical sector commands received from the file system.

17
18 **32. (Original)** The processing device as recited in Claim 23, wherein
19 the physical sector operations include read, write and error code correction
20 commands associated with the flash memory medium.

1 **33. (Currently Amended)** In a processing device that uses a flash
2 memory medium for storage of data, a method for driving the flash memory
3 medium, comprising:

4 managing rules associated with operating the flash memory medium in a
5 flash abstraction logic; and

6 issuing physical sector commands directly to the flash memory medium
7 from a flash medium logic;

8 wherein the method is flash memory medium agnostic, and wherein one of
9 the rules includes managing wear-leveling operations associated with the flash
10 memory medium by way of circular and continuous advancement of a write
11 pointer.

12
13 **34. (Original)** The method as recited in Claim 33, wherein one of the
14 rules includes maintaining data integrity of the flash memory medium.

15
16 **35. (Cancelled).**

17
18 **36. (Original)** The method as recited in Claim 33, wherein one of the
19 rules includes handling recovery of the media after a power-failure.

20
21 **37. (Original)** The method as recited in Claim 33, wherein issuing
22 physical sector commands directly to the flash memory medium comprises
23 receiving read and write commands from a file system and translating them into
24 the physical sector commands.
25

1 **38. (Original)** The method as recited in Claim 33, further comprising
2 issuing a set of programmable entry points that can be implemented by a user to
3 perform error code correction with the type of flash memory medium used in the
4 processing device.

5
6 **39. (Original)** The method as recited in Claim 33, further comprising
7 issuing a set of programmable entry points that can be optionally selected by a
8 user to interface with the type of flash memory medium used in the processing
9 device.

10
11 **40. (Original)** The method as recited in Claim 33, further comprising
12 receiving read and write commands from a file system.

13
14 **41. (Currently Amended)** One or more tangible computer-readable
15 media comprising computer-executable instructions that, when executed, perform
16 the method as recited in claim 33.

17
18 **42. (Currently Amended)** A tangible computer-readable medium
19 for a flash driver, comprising computer-executable instructions that, when
20 executed, direct the flash driver to provide an interface between a file system,
21 selected from one of a plurality of different file systems, and a flash memory
22 medium, selected from one of a plurality of different flash memory media,
23 wherein the flash driver is located remote from the flash memory medium, and
24 wherein wear-leveling of the flash memory medium is performed by way of
25 circular and continuous advancement of a write pointer.

1 **43. (Currently Amended)** A tangible computer-readable medium
2 for a flash driver, comprising computer-executable instructions that, when
3 executed, direct the flash driver to:

4 provide an interface between a file system, selected from one of a plurality
5 of different files systems, and a flash memory medium, selected from one of a
6 plurality of different flash memory media; and

7 manage a set of characteristics that are common to the plurality of different
8 flash memory media at a flash abstraction logic;

9 wherein the flash driver is flash memory medium agnostic, and wherein
10 wear-leveling of the flash memory medium is performed by way of circular and
11 continuous advancement of a write pointer.

1 **44. (Currently Amended)** A tangible computer-readable medium
2 for a flash driver, comprising computer-executable instructions that, when
3 executed, direct the flash driver to:

4 provide an interface between a file system, selected from one of a plurality
5 of different files systems, and a flash memory medium, selected from one of a
6 plurality of different flash memory media;

7 manage a set of characteristics that are common to the plurality of different
8 flash memory media at a flash abstraction logic; and

9 provide programmable entry points that can be optionally selected by a user
10 to interface with the type of flash memory medium selected;

11 wherein the flash driver is located remote from the flash memory medium
12 and the flash driver is flash memory medium agnostic, and wherein wear-leveling
13 of a flash memory medium is performed by way of circular and continuous
14 advancement of a write pointer.